

PATENT ABSTRACTS OF JAPAN

(11) Publication number : 63-016335
 (43) Date of publication of application : 23.01.1988

(51) Int.Cl.

G06F 9/46
G06F 1/00

(21) Application number : 61-159660

(71) Applicant :

HITACHI LTD
HITACHI KEIYO ENG CO LTD

(22) Date of filing : 09.07.1986

(72) Inventor :

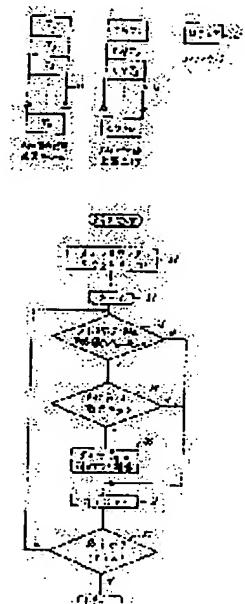
ONO RYOJI

(54) PROCESSING METHOD FOR SOFTWARE TIMER

(57) Abstract:

PURPOSE: To perform plural timer processing jobs just with a single timer by storing the value showing the specific count value with which a counter has a time-out state in a timer register mode.

CONSTITUTION: N pieces of values CNT1WN which are set a time-out state when they are coincident with the value of a main counter 13 are set in a time-out value memory area 12. A timer working state setting table 11 contains a flag showing whether a timer Tn is working or not. In a processing mode of a software timer, the counter 13 is counted up (31) with the prescribed value and it is checked (33) with N pieces of timers whether the value of the counter 13 is equal or not to the time-out value CNTn. If so, it is checked (34) whether the timer Tn of the table 11 is working or not. If so, it is reported (35) that the timer Tn has a time-out state.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

⑪ 公開特許公報 (A) 昭63-16335

⑫ Int.Cl.¹G 06 F 9/46
1/00

識別記号

310
380

厅内整理番号

E-8320-5B
7157-5B

⑬ 公開 昭和63年(1988)1月23日

審査請求 未請求 発明の数 1 (全4頁)

⑭ 発明の名称 ソフトウェアタイマの処理方法

⑮ 特願 昭61-159660

⑯ 出願 昭61(1986)7月9日

⑰ 発明者 大野 良二 千葉県習志野市東習志野7丁目1番1号 日立京葉エンジニアリング株式会社内

⑱ 出願人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

⑲ 出願人 日立京葉エンジニアリング株式会社 千葉県習志野市東習志野7丁目1番1号

⑳ 代理人 弁理士 小川 勝男 外1名

明細書

1. 発明の名称 ソフトウェアタイマの処理方法

2. 特許請求の範囲

1. タイマ機能を実現するソフトウェアにおいて、タイマの数に関係なく1個のカウンタを設け、タイマ登録時にカウンタの値がいくつになった時にタイムアウトにするかという値を記憶し、この値とカウンター一定時間ごとにカウントアップまたはカウントダウンされる値と比較し、一致している時にのみ該当タイマが起動されている状態か否かを判定し、起動されている状態であればタイムアウトとするアルゴリズムによるソフトウェアタイマの処理方法。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明はパーソナルコンピュータ等のソフトウェアによるタイマの処理方法に係り、特に通信等の時間的な応用要因のあるソフトウェアに好適なソフトウェアタイマの処理方法に関する。

〔従来の技術〕

従来のソフトウェアによるタイマの処理方法として、通常にはタイマ登録時にタイムアウト値をカウンタにセットし、一定時間ごとにこのカウンタを一定値ずつ減算して、カウンタが特定の値になった時にタイムアウトを報告する方法が用いられる。しかしこの方法はタイマの数に比例して処理時間が長くなる点について考慮されていない。

従来から通常に用いられてきたと考えられる一般的なソフトウェアタイマの処理方法の一例を第4図ないし第6図により説明する。

まず第4図は従来のソフトウェアタイマの処理方法のワークエリアの説明図である。第4図において、41はタイマ動作状態設定テーブル、42はカウンタである。いまタイマの個数の最大値Nとすると、第4図のようにN個のタイマT₁～T_Nのタイマ動作状態設定テーブルT₁～T_NおよびカウンタC_{NT1}～C_{NTN}のワークエリアを持っている。タイマ動作状態設定テーブル41は該当タイマT₁～T_Nが動作中であるか否かを記憶するのに用いられ、カウンタ42は一定時間

ごとに加算または減算されてある特定の値と比較することによりタイムアウトの判定をするのに用いられる。

第5図は従来のソフトウェアタイマの処理方法のタイマ登録フローチャートである。第5図において、いま最大値N個のタイマT₁～T_Nのうちのn番目のタイマT_nを起動するものとする。第5図のフローのようにカウンタ値を第4図のカウンタC_{NT}_nにセットし(ステップ51)、さらに第4図のタイマT_nが動作中であることを示すタイマ動作状態設定テーブル41の該当カウンタC_{NT}_nにセッティングをセットする(ステップ52)。このようにタイマ登録をしておいて、一定時間ごと起動されるタイマ処理のメインルーチンによりタイマ処理を実行する。

第6図は従来のソフトウェアタイマの処理方法のタイマ処理フローチャートである。第6図において、最大値N個の1番目～N番目のタイマT₁～T_Nのすべてをチェックするためにn番目のn

する処理は2回の比較および1回の減算となり、N個のタイマT₁～T_Nに要する処理は2N回の比較およびN回の減算となる。このようにタイマの個数の最大値Nに比例して処理時間が長くなり、多くのタイマを必要とするプログラムではタイマ処理のために性能が損なれることになる。

(発明が解決しようとする問題点)

上記従来技術は一般的なソフトウェアタイマの処理方法によって複数個のタイマを実現するさいのタイマの個数を多く必要とする場合を考慮しておらず、タイマの個数が多いほどタイマ処理時間を多く必要とするため、ソフトウェア全体の性能を低下させる問題点があった。

本発明の目的は複数個のタイマを実現するさいのタイマ処理を高速化して全体としてより高性能なソフトウェアを実現できるソフトウェアタイマの処理方法を提供するにある。

(問題点を解決するための手段)

上記目的は、タイマの数に關係なくカウンタを1個とし、タイマ登録時にカウンタの値がいくつ

を変数としてn=1～Nまでの処理を実行している。まず第6図のフローのようにn=1とし(ステップ60)、第4図のタイマ動作状態設定テーブル41によりタイマT_nが動作中か否かを判定し(ステップ61)、動作中でなければ直ちに次のタイマT_{n+1}のチェックに移る。また動作中であれば第4図のカウンタC_{NT}_nのカウンタC_{NT}_nを一定値だけカウントダウンし(ステップ62)、ある特定の値たとえば0と比較してアイムアウトが発生したか否かを判断し(ステップ63)、カウンタC_{NT}_nの値が特定の値と一致しない場合には直ちに次のタイマT_{n+1}のチェックに移る。またカウンタC_{NT}_nの値が特定の値と一致した場合にはタイムアウトが発生したものと判断して外部のルーチンにタイマのT_nのタイムアウト検出報告を行い(ステップ64)、次のタイマT_{n+1}のチェックに移る(ステップ65)。これらの処理をN回繰り返して行いn=Nとなれば終了する。(ステップ66)。

この従来方法によると、1個のタイマT_nに要

になった時にタイムアウトにするかという値を記憶し、一定時間ごとにカウントアップまたはカウントダウンされるカウンタの値と比較することにより、一致した時にのみ該当タイマが起動されている状態か否かを判定し、起動中の時にはタイムアウトとするというアルゴリズムによるソフトウェアタイマの処理方法により達成される。

(作用)

上記アルゴリズムによるソフトウェアタイマの処理方法によると、必要な処理はアイムアウトが発生していない場合の1個のタイマにつき1回の比較またはタイムアウトが発生している場合の1個のタイマにつき2回の比較に加えて1回のカウンタの加算または減算となるので、従来のアルゴリズムによるソフトウェアタイマの処理方法により必要な処理がタイムアウトが発生している場合および発生していない場合にかかわりなく1個のタイマにつき2回の比較および1回の加算または減算であったのに比較して、タイマ処理の高速化が実現できるとともにタイマの数が多い場合ほど

その効果が顕著となる。

【実施例】

以下に本発明の一実施例を第1図ないし第3図により説明する。

第1図は本発明によるソフトウエアタイマの処理方法の一実施例を示すワークエリアの説明図である。第1図において、11はタイマ動作状態設定テーブル、12はタイムアウト値記憶エリア、13はメインカウンタである。いまタイマの個数の最大値Nとすると、第1図のようにN個のタイマT₁～T_Nのタイマ動作状態設定テーブルT₁～T_Nおよびタイムアウト値記憶エリアC_{NT1}～C_{NTN}および1個のメインカウンタMC_{NT}のワークエリアを持っている。タイマ動作状態設定テーブル11は該当タイマT₁～T_Nが動作中であるか否かを記憶するのに用いられ。タイムアウト値記憶エリア12にはメインカウンタ13の値と比較して一致した時にタイムアウトとなるべき値がセットされる。メインカウンタ13は一定時間ごと加算（または減算）する1個のカウンタ

1）、つぎに最大値N個の1番目～N番目のタイマT₁～T_Nのすべてをチェックするためにn番目のnを変数としてn=1～Nまでの処理を実行している。まず第3図のフローのようにn=1とし（ステップ32）、第1図のメインカウンタ13の値とタイムアウト値記憶エリア12のエリアC_{NTn}のタイムアウト値を比較し（ステップ33）、一致していなければ直ちに次のタイマT_{n+1}のチェックに移る。また一致していればタイマアウト発生の可能性があるので、タイマ動作状態設定テーブル11のテーブルT_nによりタイマT_nが動作中か否かを判定し（ステップ34）、タイマ動作中でなければ直ちに次のタイマT_{n+1}のチェックに移る。またタイマ動作中であれば外部のルーティンにタイマT_nのタイムアウト検出報告を行い（ステップ35）、次のタイマT_{n+1}のチェックに移る（ステップ36）。これらの処理をN回繰り返して行いn=Nとなれば終了する（ステップ37）。

本方法によると、メインカウンタ13の加算の

である。

第2図は本発明によるソフトウエアタイマの処理方法の一実施例を示すタイマ登録フローチャートである。第2図において、いま最大値N個のタイマT₁～T_Nのうちのn番目のタイマT_nを起動するものとすると、第2図のフローのように第1図のメインカウンタ13の値を読み出し（ステップ21）、メインカウンタ13の値とカウントすべきカウント値を加算してカウントアウト値を算出して、カウントアウト値をカウントアウト値記憶エリア12の該当エリアC_{NTn}にセットし（ステップ22）、さらにタイマ動作状態設定テーブル11の該当テーブルT_nのタイマ動作中フラグをセットする。このようにタイマ登録をしておき、一定時間ごとに起動されるタイマ処理のメインルーチンによりタイマ処理を実行する。

第3図は本発明によるソフトウエアタイマの処理方法の一実施例を示すタイマ処理フローチャートである。第3図において、まずメインカウンタ13を一定値だけカウントアップし（ステップ3

ほかに1個のタイマT_nに要する処理は1ないし2回の比較となる。したがってN個のタイマT₁～T_Nに要する処理はNないし2N回の比較および1回の加算となる。しかも1個のタイマT_nの処理での比較のさいにメインカウンタ13の値とタイムアウト値が一致するのはごくまれであるためほとんどの場合には1回の比較ですむことになり、したがってN個のタイマT₁～T_Nに要する処理はほぼN回の比較および1回の加算となる。これは従来方法と比べると、N回の比較およびN-1回の加減算の分だけ処理量が減少していることになり、したがって処理時間もこの処理量の差による時間だけ短縮することができ、タイマの数が多くなるほどその差が大きくなつてその効果が大きくなる。

【発明の効果】

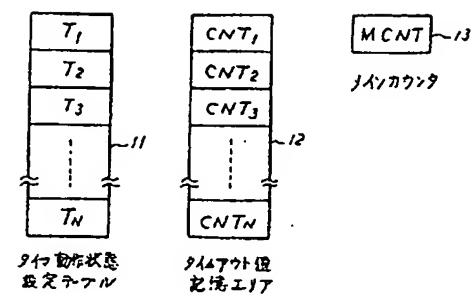
本発明によれば、ソフトウエアによるタイマ処理のアルゴリズムを改善することによってタイマ処理の高速化が実現できるので、これを用いたプログラム全体の性能向上を図れる効果がある。

4. 図面の簡単な説明

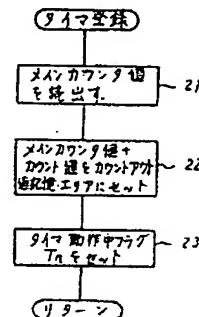
第1図は本発明によるソフトウェアタイマの処理方法の一実施例を示すワークエリア説明図、第2図は同じくタイマ登録フローチャート、第3図は同じくタイマ処理フローチャート、第4図は従来のソフトウェアタイマの処理方法を例示するワークエリア説明図、第5図は同じくタイマ登録フローチャート、第6図は同じくタイマ処理フローチャートである。

1 1 …タイマ動作状態設定テーブル、1 2 …タイマアウト値記憶エリア、1 3 …メインカウンタ

第1図

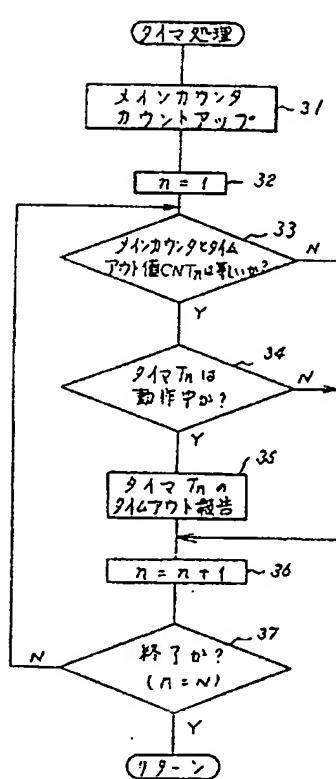


第2図

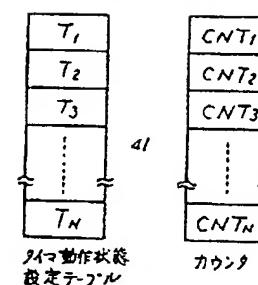


代理人弁理士 小川勝男

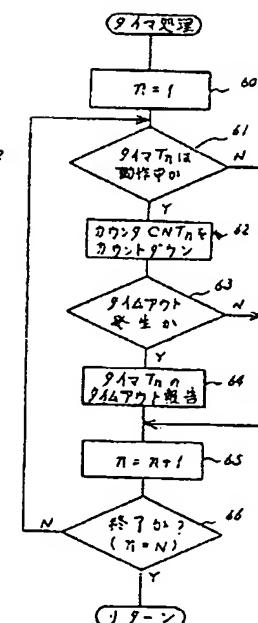
第3図



第4図



第6図



第5図

